Report #7
PD-Internship
Lerner Paul
31/07/2018

As planned, I'm currently working on transfer learning techniques to overcome the lack of available data and the insuccess of data augmentation.

Contents

1 Auto-encoder

In report #4 I described the works of Dai et al. and Malhotra et al. who both use an auto-encoder (AE) in order to pre-train a model which task is to classify a sequence.
I liked this approach because, as AEs are unsupervised they allow for training on a large unsupervised database. Moreover, I'm not sure which supervised learning task would we suitable for transfer learning in our case, maybe mode (i.e. text/drawing) detection ?
However, I talked about it with Leo a while back and he told me that AEs are not suitable for online handwriting as the data has a small dimension compared to, e.g. word embeddings like in Dai et al. work. **This results in the AE learning the identity function**, regardless of the input, early coding experiments suggests that as well :
I pre-trained a Convolutional AE on the Iamon-do database (cf report #5). At the time I used an open-source-python code to load the data but it loaded only the measures of x and y coordinates. However, I was very surprised that, after training the model on Iamondo, not

only the model is able to reconstruct the time series of Iamondo, PaHaW but it's able to reconstruct the time series of PaHaW when feeding only, e.g. elevation and pressure measures from PaHaW ! Although this demonstrates the learning power of neural networks, it suggests that the AE doesn't learn any measure-specific features. However, since the AE is still able to reduce a time series of shape (288, 2) into an encoding vector of 192 I thought the encoding representation might contain some useful information. Thus I tried to train a model by adding a FC layer with sigmoid activation after the encoder part of the AE. I tried to :

1. freeze the AE weights to fully evaluate the quality of the learned encoding
2. initialize the model weights with the AE's but then fine-tune it
3. initialize the model weights without transfer learning, with a uniform distribution (as usual)

As Iamondo contains only on-paper strokes I evaluated the model on the on-paper strokes of PaHaW. On the I task, the model achieves better results without transfer learning than with it. In future experiments, if this "transfer learning test" is passed, we should also try to :

1. train the model using both on-paper and in-air strokes (since we've seen in the latest experiments that CNN-Three is able to learn from both at the same time).
2. train the model on all measures by initializing weights of other measures uniformally

1.1 KL divergence

To overcome **AE learning the identity function**, Fayyaz et al. use Kullback-Leibler divergence as a loss function for training the AE :

> *"This brings us a network with the ability of representing raw data with learned feature without any guarantee of having sparse represented features, which plays a key role in classification task. In order to learn features that are more effective and having a sparser dataset of represented features, the sparsity constraint can impose on the autoencoder network".* — Fayyaz et al.

The sparsity constraint constrains the neurons to be inactive most of the time (i.e. value close to 0). Therefore, if I understood correctly, we need to have a Sigmoid activation (the decoder doesn't have any activation function after the last layer in the current implementation), thus scaling the data between 0 and 1 (Ng, 2011).

1.2 Denoising AE

> *"Without any additional constraints, conventional auto-encoders learn the identity mapping. This problem can be circumvented by using a probabilistic RBM approach, or sparse coding, or denoising auto-encoders (DAs) trying to reconstruct noisy inputs."* — Masci et al.

Masci et al. show that using denoising AE or pooling layers is essential to learn "biologically plausible features". In practice however,

1.3 High dimensional data

In paper where the data is highly dimensional (e.g. video, word embeddings), author don't mention the problem of the identity function. See Baccouche et al., Dai et al., Le Guennec et al. Malhotra et al.

2 Mode detection

The principal use of the IAMonDo-database is mode detection (i.e. handwriting / drawing). In report #5 I described the work of Indermuhle et al. who treat this as a sequence labeling task (they achieve 0.9701 accuracy). I thought that our model could be pre-trained to this task before being fine-tuned in PaHaW.
CNN-Three (cf. report on the CNN) achieves 0.7460 accuracy using the same training and test sets as Indermuhle et al. I arbitrarily set the number of epochs at 50 as the training curve is very stable. As I'm only interested in the transfer learning task, I don't combine the predictions of the model, meaning that I present here the stroke-accuracy. I achieved better results (0.7839 accuracy) using a more classical model than CNN-Three : CNN-Five which has a growing number of filters on a $\log_2$ scale going from 16 to 256, a decreasing kernel size going from 128 to 8 and a pooling kernel of 2 or 3 for every layer.
The first transfer learning experiments provide encouraging results : on the l task, the model achieves better results after fine-tuning than with uniform initialization, see table below :

Table. 10 CV accuracy of CNN-Five on the l task, w.r.t. transfer learning

uniform initialization	*pre-trained conv layers*	*frozen conv layers*
0.65 (± 0.12)	**0.74** (± 0.10)	0.56 (± 0.16)

However, when fine-tuning CNN-Five on all tasks, it stills fall below CNN-Three after majority voting from a large margin. This may be explained because we have to downsample PaHaW from 150 Hz to 75 Hz as IAMonDo-database is only 75 Hz, so information might be lost during downsampling. Moreover, since IAMonDo-database only provides for on-paper strokes and for the spatial coordinates, pressure and time stamp measure ; we can't use neither the in-air strokes nor the tilt and elevation measures. Moreover, since CNN-Five was

trained on IAMonDo which is padded at 1050 timesteps, I had to trim some of the strokes (especially the spirals).

Conclusion - Todo List

These first experiments provide encouraging results but I'm not sure I'll have time to dig into it more in depth before the end of my internship. I leave tomorrow for holiday until August 12th and I won't be able to work until August 10th. When I come back I'll have to :

1. Write my internship report (~ 30 pages) : deadline September 6th
2. Prepare the slides and my presentation for the defense (~ 20 min. + 5 min. of questions) : deadline September 12th
3. Prepare the poster if my paper is accepted at JDSE : deadline September 12th
4. Write a full-length paper for IEEE BIBM[1] : deadline August 17th

References

Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., & Baskurt, A. (2012, September). Spatio-Temporal Convolutional Sparse Auto-Encoder for Sequence Classification. In BMVC (pp. 1-12).

Chen, M., Shi, X., Zhang, Y., Wu, D., & Guizani, M. (2017). Deep features learning for medical image analysis with convolutional autoencoder neural network. IEEE Transactions on Big Data.

Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. In Advances in neural information processing systems (pp. 3079-3087).

Fayyaz, M., Hajizadeh_Saffar, M., Sabokrou, M., & Fathy, M. (2015). Feature representation for online signature verification. arXiv preprint arXiv:1505.08153.

Indermühle, E., Frinken, V., & Bunke, H. (2012, September). Mode detection in online handwritten documents using BLSTM neural networks. In 2012 International Conference on Frontiers in Handwriting Recognition (pp. 302-307). IEEE.

Le Guennec, A., Malinowski, S., & Tavenard, R. (2016, September). Data augmentation for time series classification using convolutional neural networks.

[1] http://ieeebibm.org/BIBM2019/index.html

Malhotra, P., TV, V., Vig, L., Agarwal, P., & Shroff, G. (2017). TimeNet: Pre-trained deep recurrent neural network for time series classification. arXiv preprint arXiv:1706.08838.

Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J. (2011, June). Stacked convolutional auto-encoders for hierarchical feature extraction. In International Conference on Artificial Neural Networks (pp. 52-59). Springer, Berlin, Heidelberg.

Ng, A. (2011). Sparse autoencoder. *CS294A Lecture notes*, *72*(2011), 1-19.